M1_lab

You are logged in.

1. List all of your variables

```
[user@c1~]$ set | more
```

2. List all exported variables

```
[user@c1~]$ env | more
```

3. Create a variable called myvar and fill it with your name
   Start a new bash shell
   What is the content of myvar?

```
[user@c1~]$ myvar=peter
[user@c1~]$ bash
[user@c1~]$ echo $myvar

[user@c1~]$
```

4. Exit the shell
   Make sure the content of myvar is also available in a child shell

```
[user@c1~]$ export myvar
[user@c1~]$ bash
[user@c1~]$ echo $myvar
peter
[user@c1~]$
```

5. When you start a new shell, make sure that the current date is
   displayed when the shell gets started.

```
[user@c1~]$ echo date >> ~/.bashrc
```

6. How can you query the returncode of the last command that was executed?

```
[user@c1~]$ echo $?
```

7. Make sure that the following output will be printed:

    This product will cost $10.

```
[user@c1~]$ echo This product will cost \$10
```

8. Change your primary system prompt. It should only display your username.

```
[user@c1~]$ PS1="$LOGNAME "
```

9. Can you delete a variable?

```
[user@c1~]$ number=1
[user@c1~]$ echo $number
1
[user@c1~]$ unset number
[user@c1~]$ echo $number

[user@c1~]$
```

10. List your aliases.

```
[user@c1~]$ alias | more
```

11. Create an alias that will list all files sorted by time.
    The alias name should be: listall

```
[user@c1~]$ ls -t
labfiles  scripts  vi  textfiles
```

12. Use output redirection to create a file with all
filenames of your login directory.

```
[user@c1~]$ ls ~/ > ~/allfiles
```

13. Use double output redirection to add all filenames
of the /root directory to the file you created
in exercise 1.

```
[user@c1~]$  ls /root >> ~/allfiles
```

14. As 'centos', run the 'find' command to search
all files under "/". Redirect all filenames to
$HOME/flist. Redirect all errors to $HOME/errors.

```
[user@c1~]$ find / > ~/flist 2>~/errors
```
or
```
[user@c1~]$ find / > $HOME/flist $HOME>~/errors
```

15. Use command-line piping to find out the number
of users logged in to your system.

```
[user@c1~]$ who | wc -l
2
```

16. What is the result of the following command:
ls | wc -l > ls

```
[user@c1~]$ ls | wc -l > ls
[user@c1~]$ cat ls
9
```

17. Is there a file called data2k.c in your environment?

```
[user@c1~]$ find . -name data2k.c
./labfiles/data2k.c
```

What does the following command do:
gcc ./labfiles/data2k.c -o ./labfiles/data2k || vi data2k.c

18. What does the following command do:
find / -iname "*.txt" | tee findit.out

19. Use the 'cut' command to list the first field of the /etc/hosts file.

```
[user@c1~]$ cut -d" " -f1 /etc/hosts
```

20. Use the 'tail' command to list all lines in
/etc/hosts, but skip the first four lines.

```
[user@c1~]$ tail -n +5 /etc/hosts
```
(**note:** tail starts with 0. So to skip the first 4 lines, you have to really skip 5.)

21. Use the 'strings' command to view all errors
that the 'ps' command can generate that start
with "list".

```
[user@c1~]$ strings /usr/bin/ps | grep -i "^list"
```

22. When you run the '`cat /etc/ssh/sshd_config`' command,
you see there are empty lines.
Use the 'sed' command to delete the empty lines
in the output.

```
[user@c1~]$ sudo cat /etc/ssh/sshd_config | sed /^$/d
```

23. Use three different commands to display the hostname of your machine.

```
[user@c1~]$ uname -n
[user@c1~]$ hostname
[user@c1~]$ echo $HOSTNAME
```

24. What does the **help** command do?

25. Use the **file** command to determine all file types in your login directory.

```
[user@c1~]$ file ~/*
```

26. Create a file with all absolute pathnames on your machine.
    Then split that file into smaller files with 100 lines each.
    The prefix of every file should be “part”.

```
[user@c1~]$ find / > ~/allfiles
[user@c1~]$ split -l 100 allfiles part
```

27. Remove all files that start with “part”.
    Repeat task 26, but now using a single command line.
    (the “-” character represents standard input)

```
[user@c1~]$ rm part*
[user@c1~]$ find  / | split -l 100 - part
```

28. Display the last 5 lines of the **/var/log/messages** file

```
[user@c1~]$ tail -5 /var/log/messages
tail: cannot open '/var/log/messages' for reading: Permission denied
[user@c1~]$ sudo tail -5 /var/log/messages
```

29. Use the **cut** command to display only the ip addresses from the /etc/hosts file.
    Use the **delimiter** option to set the field separator to a space character.

```
[user@c1~]$ cut -d" " -f1 /etc/hosts
```